Markowitz efficient frontier states investors should consider multiple securities in a portfolio rather than individually. A portfolio that contains combination of securities with low correlation can benefit from a diversification effect. Meaning investors can optimize their return without assuming additional risk. Markowitz

In [2]:
```
import numpy as np
import pandas as pd
from pandas_datareader import data as web
import matplotlib.pyplot as plt
from scipy import stats
%matplotlib inline
```

WE will download the data on PG stock and ^GSPC

In [57]:
```
tickers = ["PG","^GSPC"]
data = pd.DataFrame()
for t in tickers:
    data[t] = web.DataReader(t, data_source = "yahoo", start = "2012-1-
1", end = "2018-12-31")["Adj Close"]
```

In [58]:
```
#normalize the data
(data/data.iloc[0]*100).plot(figsize = (16,8))
plt.show()
```


Calculate the daily change, returns of both securties

In [59]:
```
simple_returns = (data/data.shift(1)) - 1
```

In [60]:
```
#we will check if the data matches and have equal values - > 2494 PG and
2494 ^GSPC
simple_returns.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1760 entries, 2012-01-03 to 2018-12-31
Data columns (total 2 columns):
PG       1759 non-null float64
^GSPC    1759 non-null float64
dtypes: float64(2)
memory usage: 41.2 KB
```

In [ ]:

In [61]:
```
#check the tail end of the data to check for most current date
simple_returns.tail()
```

Out[61]:

| Date | PG | ^GSPC |
|---|---|---|
| 2018-12-24 | -0.039683 | -0.027112 |
| 2018-12-26 | 0.031250 | 0.049594 |
| 2018-12-27 | 0.021423 | 0.008563 |
| 2018-12-28 | -0.009128 | -0.001242 |
| 2018-12-31 | 0.008116 | 0.008492 |

In [62]:
```
simple_returns.cov() * 250
```

Out[62]:

| | PG | ^GSPC |
|---|---|---|
| PG | 0.021513 | 0.009302 |
| ^GSPC | 0.009302 | 0.016424 |

In [63]:
```
#the correlation between PF and ^GSPC is positive but low so the portfol
io should benefit from
#markowitz diversification effect
simple_returns.corr()
```

Out[63]:

| | PG | ^GSPC |
|---|---|---|
| PG | 1.000000 | 0.494857 |
| ^GSPC | 0.494857 | 1.000000 |

In [64]:
```
# portfolio optimization -> We will need the count of securities in the
 portfolio
port_asset = len(tickers)
print(f"The number of securties in the portfolio is {port_asset}")
```

```
The number of securties in the portfolio is 2
```

WE will need the expected returns and the volatility to simulate a mean variance combination with 1000 simulations. WE are considering 1000 combinations of the same 2 assets of their weight values not 1000 different investments.

In [ ]:
```

```

In [65]:
```
#Bellow we will run a simulation of 1000 differenct portfolio that conta
in PG and ^GSPC to test Markowitz theory.
#This Will provide us with both 1000 different expected returns and 1000
volatility values

portfolio_returns = []
portfolio_volatilities = []
weights1 = []
weights2 = []

for x in range(1000):
    weights = np.random.random(port_asset)
    weights[0] = weights[0]/np.sum(weights)
    weights[1] = weights[1]/np.sum(weights)
    weights /= np.sum(weights)
    weights1.append(weights[0])
    weights2.append(weights[1])
    portfolio_returns.append(np.sum(weights * simple_returns.mean()) * 2
50)
    portfolio_volatilities.append(np.sqrt(np.dot(weights.T, np.dot(simpl
e_returns.cov() * 250, weights))))

    # we will need to convert the volatilities and and the ezpected retu
rns into a numpy array
port_Returns = np.array(portfolio_returns)
port_Vol = np.array(portfolio_volatilities)
```

In [66]:
```
df2 = pd.DataFrame(port_Returns, columns=["Returns"])
df2["Risk"] = port_Vol
df2["Weight  PG"] = weights1
df2["Weight ^GSPC"] = weights2
```

In [67]:
```
df2.tail()
```

Out[67]:

| | Returns | Risk | Weight PG | Weight ^GSPC |
|---|---|---|---|---|
| **995** | 0.091047 | 0.132618 | 0.810997 | 0.189003 |
| **996** | 0.090460 | 0.135047 | 0.847484 | 0.152516 |
| **997** | 0.088821 | 0.142562 | 0.949287 | 0.050713 |
| **998** | 0.093337 | 0.124676 | 0.668727 | 0.331273 |
| **999** | 0.097110 | 0.117831 | 0.434282 | 0.565718 |

In [68]:
```
portfolios = pd.DataFrame({"Volatility": port_Vol, "Returns": port_Retur
ns})
```

In [69]:
```
portfolios.head()
```

Out[69]:

| | Volatility | Returns |
|---|---|---|
| **0** | 0.117623 | 0.097487 |
| **1** | 0.134139 | 0.090675 |
| **2** | 0.119914 | 0.095384 |
| **3** | 0.120421 | 0.101235 |
| **4** | 0.117480 | 0.098298 |

In [70]:
```
portfolios.tail()
```

Out[70]:

| | Volatility | Returns |
|---|---|---|
| **995** | 0.132618 | 0.091047 |
| **996** | 0.135047 | 0.090460 |
| **997** | 0.142562 | 0.088821 |
| **998** | 0.124676 | 0.093337 |
| **999** | 0.117831 | 0.097110 |

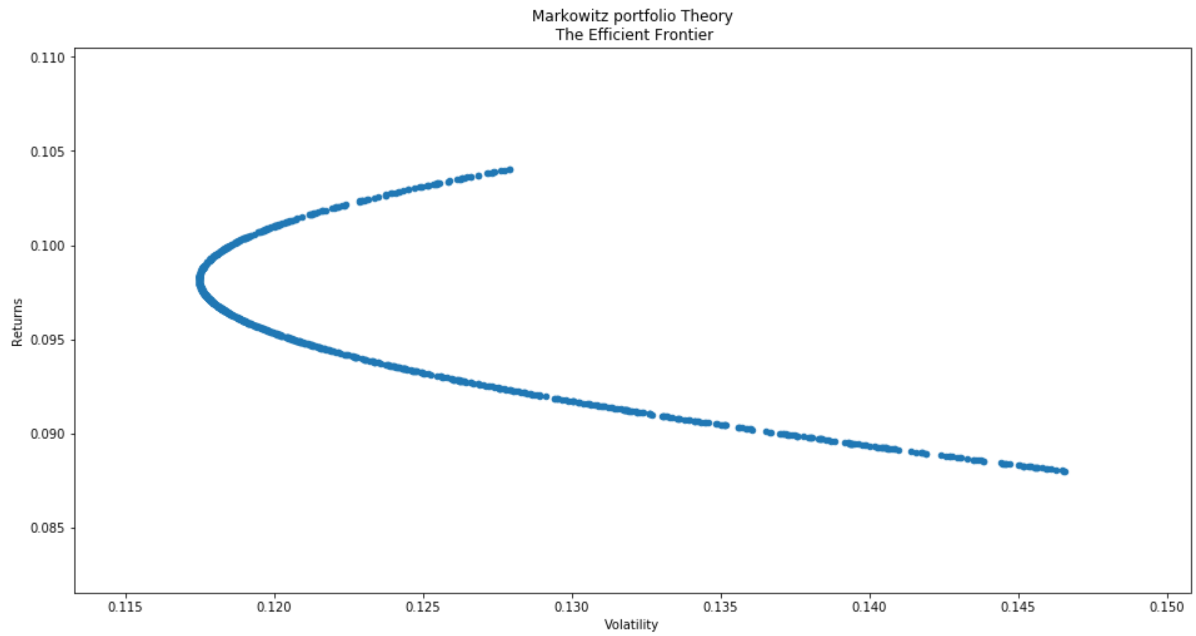
In [71]:
```
portfolios["Volatility"].min()
```

Out[71]:
```
0.11747431618324211
```

In [72]:
```
portfolios.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
Volatility    1000 non-null float64
Returns       1000 non-null float64
dtypes: float64(2)
memory usage: 15.7 KB
```

In [73]:
```
portfolios.plot(x ="Volatility", y ="Returns", kind = "scatter", figsize
= (16,8))
plt.title("Markowitz portfolio Theory\n The Efficient Frontier")
plt.show()
```




THe above graph shows a set of 1000 portfolios of different weights containing PG & ^GSPC, and displays the typical shape of Markowitz efficient portfolio. There are a set of efficient portfoilios that can provide a higher rate of return for the same or lower risk. The starting point is the minimum variance portfolio.

In [ ]:

In [ ]:

In [ ]: